



HTTPantry

Rowen Felt and Zebediah Millslagle
CS701 - Spring 2019



Background

- During Development, testing a web app can result in a large number of RESTful API requests
- Calling these requests can be very expensive, both in terms of money and network traffic
- Also may be the case that a RESTful API is not set up yet, but developers might still want to use it in test-driven development (with the knowledge of what the API *should* return)



Our Goal

Build a configurable caching tool that:

- Stores responses dynamically, resolving unstored responses
- Can be persistent between sessions
- Can be installed and deployed in under a minute
- Allows for custom responses
- Can be easily configured
- Can handle both HTTP and HTTPS requests



Our Process - Initial Ideas

- Intercept requests at IP layer
 - Modify /etc/hosts file and redirect to local HTTP server
 - Bad idea for many, many reasons
- Write a wrapper for a library that already resolves requests
 - Not very flexible
 - Need to provide access to other functions in library

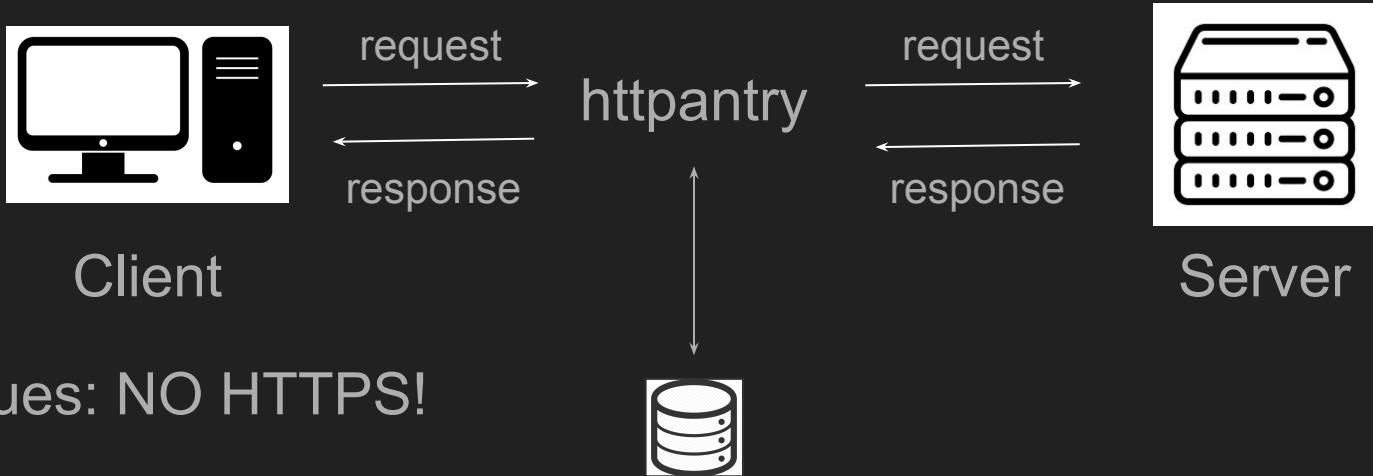
Our Process - Proxy Server



- Variable API routes

```
16 #catch all URL
17 @app.route('/', defaults={'path': ''})
18 @app.route('/<path:path>', methods=['GET', 'POST', 'PUT', 'DELETE'])
```

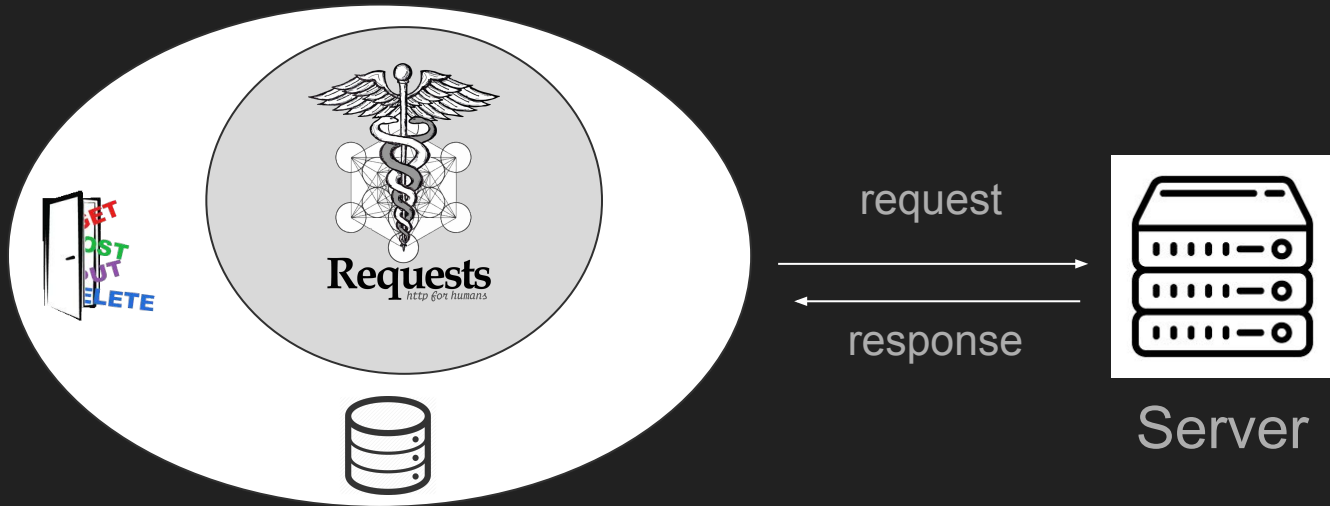
- Man in the middle



- Issues: NO HTTPS!

Our Process - Requests Wrapper

- Wrapper around the Requests library for HTTPS
- Limited to Python Requests library, but very fast and easy to use



Tech Stack



ConfigParser



Ease of Use

- Easy installation using pip
- User needs to only add a few lines to existing code
- Proxy can be deployed from the command line

```
import httppantry as requests
```

```
proxy_urls = {
    'http': 'http://localhost:5000'
}
```

```
[zeb@zebediahs-mbp:~]$ httppantry server
* Serving Flask app "httppantry.http_proxy_server" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 188-677-605
```




Customizable Features

- Directory created wherever process is run
- Allows for configuration of proxy server, database, timeouts, and custom responses
- Implemented with ConfigParser

Folder	__httpantry_cache__	Today at 1:23 PM	--	Folder
TextEdit	config	Today at 1:23 PM	280 bytes	TextEdit
JSON	custom_responses.json	Today at 1:23 PM	458 bytes	JSON
Document	proxy_database.db	Today at 1:23 PM	29 KB	Document

[GENERAL]

port_number = 5000

persistence = True

uncached_apis = <http://httpbin.org/image/jpeg>

[CUSTOM_RESPONSES]

use_custom_responses = True

file_name = __httpantry_cache__/custom_responses.json

[TIMEOUT]

days = 0

hours = 1

minutes = 0

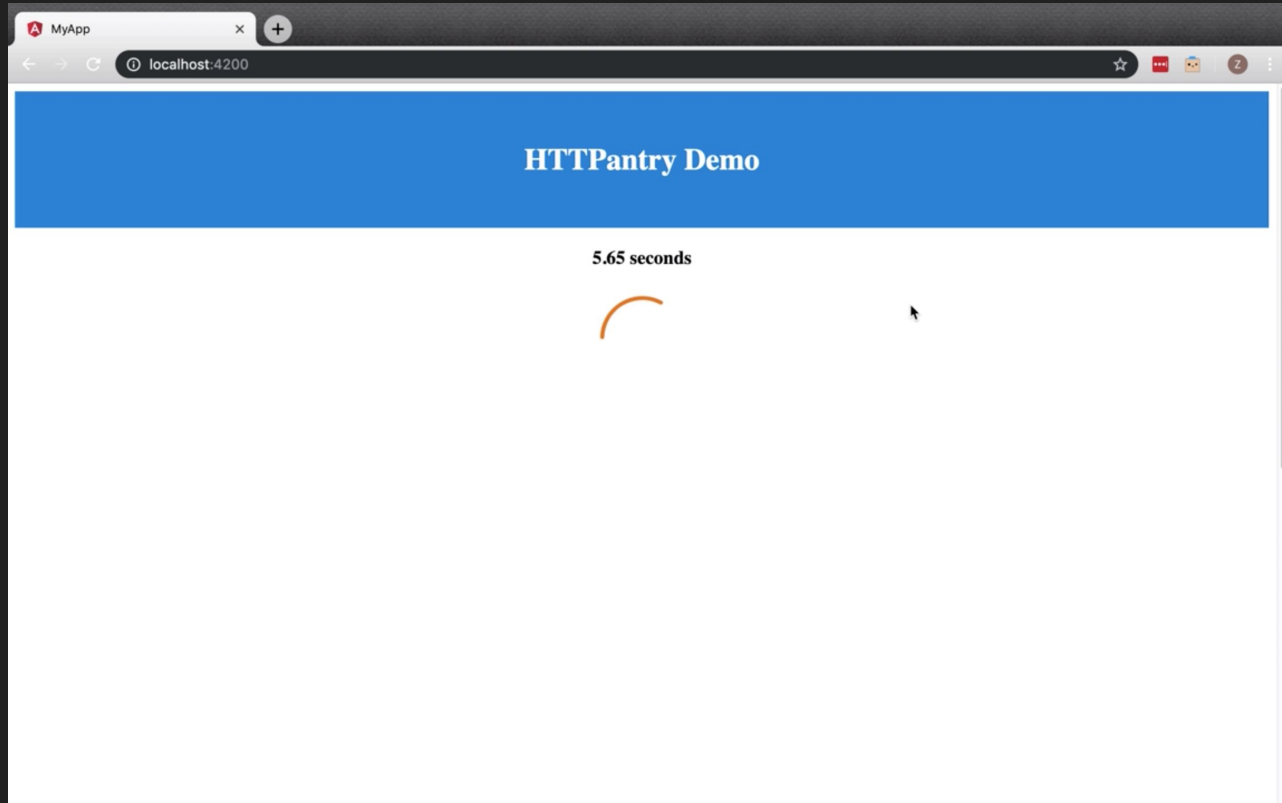
seconds = 0

milliseconds = 0

Demo



Backup Demo





Conclusion

- Accomplished our original goals
 - httpantry can be installed, configured, and run in under a minute
 - Both implementations allow a user to cache responses without substantial changes to their codebase
- Future work
 - Proxy encryption and security
 - HTTPS in proxy
 - More flexible wrapper

Conclusion

Questions?

Comments?

Concerns?



Image Sources

<http://www.cielaustral.com/galerie/photo95f.jpg>

<https://www.visualpharm.com/free-icons/server-595b40b75ba036ed117d8edd>

<https://www.vectorstock.com/royalty-free-vector/computer-icon-isolated-on-white-pc-symbol-vector-19668605>

<http://flask.pocoo.org/>

